

REMARKS

Claims 1-16 are all the claims pending in the application.

I. Summary of the Office Action

Claims 1-16 are all the claims pending in the application. In this third Office Action, the Examiner maintained the previous grounds of rejections. Claims 1-16 presently stand rejected under 35 U.S.C. § 103(a).

II. Prior Art Rejection

Claims 1-16 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Publication No. 2002/0161915 A1 to Crow et al. (hereinafter “Crow”) and U.S. Patent Publication No. 2003/0069973 A1 to Ganesan et al. (hereinafter “Ganesan”) in view of U.S. Patent Publication No. 2004/0037302 A1 to Varma et al. (hereinafter “Varma”) and in further view of U.S. Patent Publication No. 2002/0095512 A1 to Rana et al. (hereinafter “Rana”). Applicant respectfully traverses these grounds of rejection at least in view of the following exemplary comments.

Claim 1 recites *inter alia* “if the received packet is the first fragment packet,
...comparing the result of the looked-up fragment ID with each list of a fragment look-up table
into which the results of fragment looked-ups for other received packets are entered, to
determine if there is a corresponding list; searching an index indicating one of the protocol
processors and corresponding to the tunnel ID of the received packet from a tunnel ID look-up
table, and if the list corresponding to the result of the looked-up fragment ID exists in the

fragment look-up table, entering the index into the corresponding list of the fragment look-up table.”

In an exemplary embodiment, subsequent fragment packets that arrive before the first fragment packet are temporarily stored in the fragment buffer with a corresponding fragment ID until the first packet is received. That is, the subsequent fragment packets do not include the tunnel ID. Accordingly, after the first packet has been received, the system looks-up that tunnel ID and searches the index corresponding to that tunnel ID to attach the index as a tag to the subsequent fragment packets. In other words, when the first packet is received, a valid index is input into a corresponding list of the fragment look-up table.

Specifically, in an exemplary embodiment, the system looks-up the fragment ID and searches the fragment ID to determine whether there is a list corresponding to this fragment ID (*i.e.*, if there are subsequent fragment packets that were received before this first packet). The packets are then transmitted to a protocol processor without reassembly *i.e.*, since each fragment now has a tag *i.e.*, a valid index.

It will be appreciated that the foregoing remarks relate to the invention in a general sense, the remarks are not necessarily limitative of any claims and are intended only to help the Examiner better understand the distinguishing aspects of the claim mentioned above.

The Examiner acknowledges that Crow and Ganesan do not disclose or suggest the above-noted unique features of claim 1. The Examiner, however, alleges that Varma cures the above-identified deficiencies of Crow and Ganesan. Specifically, in response to Applicant's arguments, the Examiner alleges that Varma discloses searching a corresponding list in order to update various pointer values for the linked list and maintain fragment count. The Examiner further alleges that the fragment ID of each received fragment has to be compared with the

fragment ID in the control memory to be able to update corresponding fragment count and the pointers in the link memory (*see* pages 29-30 of the Office Action). The Examiner's position is not understood.

Applicant respectfully maintains that Varma does not disclose or suggest searching if a corresponding list exists when the first fragment is received. In Varma, it is clear that when the data is received for the empty queue, no searches are performed. That is, if a queue is empty, there is no need to search for an existing list.

Furthermore, in Varma, there is no disclosure or even remote suggestion that the first packet fragment is received after the other packet fragments. However, even assuming *arguendo* that the first packet fragment is one of the subsequently received fragments, there is no need to search for an existing list (queue) as one clearly exists. In other words, updating pointers in the list cannot disclose or suggest searching to determine if a corresponding list exists or is present. In Varma, there is no disclosure or suggestion of searching to determine whether or not the queue exists.

In short, Varma does not disclose or suggest if the data is the first data, searching to determine if there is a corresponding list of other received packet fragments. Varma does not disclose or even remotely suggest determining if there is a corresponding list when the first data block is received.

In addition, in Varma, when the first block is received, there is no disclosure or even remote suggestion to look for other received packets at least because the data received is the first block that is assigned the head and tail pointers and as such there are no other data blocks yet. In Varma, there is no disclosure or suggestion of fragment IDs. Accordingly, the Examiner's rebuttal is technically inaccurate as it is not based on Varma's disclosure.

In short, Varma clearly does not disclose or suggest that if the first fragment packet is received, comparing fragment ID to determine if there exists a corresponding list with other received packets for this fragment ID. Rana does not cure the above-identified deficiencies of Crow, Ganesan, and Varma.

With respect to the Rana reference, in response to Applicant's arguments, the Examiner alleges that session ID corresponds to the index set forth in claim 1 (*see* pages 30-31 of the Office Action). Applicant respectfully disagrees.

Rana discloses the packet assembler uses unique fields in the data packet to access a session ID, which is used to associate the data packet with a particular traffic flow over the network. The session ID allows each data packet to be assigned to a traffic flow so sequence numbers can be used to anticipate the next data packet and out of order packets can be identified. Out of order packets are sent to a reordering unit, which reorders the data packets by modifying links to the packet memory (§ 11). Rana discloses that a session ID is assigned to the first data packet for a new session, and each subsequent packet in the session is associated with that session ID.

Nowhere, however, does Rana disclose entering an index into the corresponding list. In Rana, a session ID is simply assigned to the first data packet and subsequent data packets. In Rana, there is no disclosure or suggest of the index being valid (when the first packet was already received) and not valid (when the first packet is not yet received). In other words, in Rana, there is no disclosure or suggestion of entering a session ID. In Rana, the session ID is automatically assigned to each packet.

That is, the session ID is not assigned when the list corresponding to the result of the looked-up fragment ID exists in the fragment look-up table. In Rana, the session ID is assigned

to the first and subsequent packets without checking whether a particular condition (*i.e.*, if the list corresponding to the result of the looked-up fragment ID exists in the fragment look-up table) is valid. In short, Rana does not disclose or even remotely suggest assigning an index to the packets stored in the corresponding list when the first packet is received. In Rana, there is no disclosure or suggestion of the index of subsequent packets being invalid prior to receipt of the first packet.

Therefore, “if the received packet is the first fragment packet, ...comparing the result of the looked-up fragment ID with each list of a fragment look-up table into which the results of fragment looked-ups for other received packets are entered, to determine if there is a corresponding list; searching an index indicating one of the protocol processors and corresponding to the tunnel ID of the received packet from a tunnel ID look-up table, and if the list corresponding to the result of the looked-up fragment ID exists in the fragment look-up table, entering the index into the corresponding list of the fragment look-up table,” as set forth in claim 1 is not disclosed by the combined disclosure of Crow, Ganesan, Varma, and Rana. Together, the combined teachings of these references would not have and could not have rendered obvious the unique features of claim 1. For at least these exemplary reasons, claim 1 is patentable over Crow in view of Ganesan, Varma, and Rana. Claims 2-6 and 12-16 are patentable at least by virtue of their dependency on claim 1.

In addition, dependent claim 13 recites: “wherein, if the received fragment packet is determined to be the first fragment packet and the first list is found in the look-up table, editing the list to update the index and searching the fragment buffer for the other received fragment packets and transmitting the found other received fragments based on the updated index of the first list without assembling the fragment packets to form the datagram.” Crow, Ganesan,

Varma, and Rana do not disclose or suggest editing the list to update the index when the received packet is determined to be the first packet. On the contrary, the session ID in Rana is not updated. It is assigned and does not change when first packet is received. For at least these additional exemplary reasons, claim 13 is patentable over the prior art of record.


Next, independent claim 7 recite features similar to, although not necessarily coextensive with, the features argued above with respect to claim 1. Therefore, arguments presented with respect to claim 1 apply with equal force here. For at least substantially analogous exemplary reasons, therefore, independent claim 7 is patentable over Crow and Ganesan. Claims 8-11 are patentable at least by virtue of their dependency on claim 7.

III. Conclusion

In view of the above, reconsideration and allowance of this application are now believed to be in order, and such actions are hereby solicited. If any points remain in issue which the Examiner feels may be best resolved through a personal or telephone interview, the Examiner is kindly invited to contact the undersigned attorney at the telephone number listed below.

The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,



Nataliya Dvorson
Registration No. 56,616

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE

23373

CUSTOMER NUMBER

Date: October 6, 2008